
Vers l'utilisation de Bluetooth pour la commande à distance de robots mobiles

Thierry Val* — **Philippe Fraisse**** — **David Andreu****

** Equipe ICARE (EA 3050)
1, Place Georges Brassens
F-31703 Blagnac Cedex
val@iut-blagnac.fr*

*** Laboratoire d'Informatique Robotique Micro-électronique (UMR5593)
161, rue Ada
F-34392 Montpellier Cedex 5
fraisse@lirmm.fr, andreu@lirmm.fr*

RÉSUMÉ. Ce papier présente l'analyse et la conception d'un système de communication sans fil basé sur la technologie Bluetooth destinée à la commande à distance de manipulateurs mobiles. Une présentation de la technologie Bluetooth et de ses mécanismes de transmission sont présents ainsi qu'une caractérisation du domaine de fonctionnement de la commande à distance d'un manipulateur mobile en présence de retards.

ABSTRACT. This paper proposes a solution to control some remote mobile manipulators using Bluetooth technology. Some Bluetooth specifications and performances are presented as well as experimental results. A remote mobile manipulator control case is theoretically studied including the stability analysis with time delay. Some simulations results prove the robustness of this control law versus time delay.

MOTS-CLÉS : Bluetooth, commande robuste, temps réel, sans fil, QoS.

KEYWORDS: Bluetooth, Robust Control, Real-Time, Wireless, QoS.

1. Introduction

C'est dans le cadre de la collaboration scientifique L2I [Fraisie *et al.*, 2000] que s'inscrivent les premiers résultats de cette étude. L'objectif est d'identifier et analyser les différents problèmes scientifiques et verrous technologiques relatifs à la mise en œuvre d'un système de commande à distance de manipulateurs mobiles *via* un réseau sans fil. Un site expérimental composé de trois robots communicants est en cours de développement. On peut voir sur la figure 1 le premier prototype de robot mobile qui sera à terme équipé d'un minibras manipulateur et de la carte fille Bluetooth dotée du composant Ericsson.

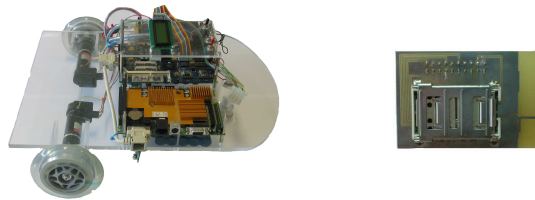


Figure 1. Robot mobile équipé d'un module Bluetooth

Ce papier présente l'analyse et la conception d'un système de communication sans fil basé sur la technologie Bluetooth destinée à la commande à distance de robots manipulateurs mobiles. Les robots mobiles utilisés dans des applications de surveillance ou de maintenance ont des contraintes mécaniques (petites tailles), énergétiques (accumulateur de faibles capacités) mais aussi d'évolutivité (modifications des fonctionnalités) qui induisent une restriction forte sur l'électronique et l'informatique embarquées. Ces contraintes imposent de minimiser l'informatique embarquée au strict nécessaire et de déporter vers une station fixe les applications de décisions et de traitement des informations de certains capteurs. Cette délocalisation des tâches nécessite un lien de communication sans fil entre le mobile et une base connectée à la station fixe. Ce lien est de plus nécessaire pour une prise en main à distance par un opérateur lors de tâches d'exceptions.

Une présentation des différents cas d'exploitation met en place le contexte de travail. Cette hypothèse de travail sur la délocalisation de boucle de commande de haut niveau (espace cartésien, espace opérationnel, téléopération) nous amène à gérer le lien de communication sans fil dans la boucle de commande, soit dans le cas d'une téléopération soit dans celui d'une boucle de commande distante. Cette gestion nécessite la prise en compte dans les lois de commande bas niveau des retards de transmission. Aussi est-il important de caractériser le domaine de fonctionnement des systèmes commande en fonction de ces retards. Une présentation de Bluetooth permet de cerner les limitations et les performances que l'on peut attendre d'une telle technologie.

2. Commande distante *via* un réseau sans fil

2.1. Architecture des lois de commande

La commande coordonnée de robots manipulateurs mobiles pour le transport d'objet lourd et encombrant (cf. section 3.3) nécessite la connaissance au niveau du robot maître de la position et l'orientation de chacun des robots (P_{ri}, V_{ri}) esclaves afin de calculer les positions désirés à l'instant suivant (cf. figure 2), mais aussi l'ensemble des efforts mesurés sur les bras manipulateurs pendant le mouvement. Le robot maître pourra, selon la tâche effectuée, réaliser la coordination bas niveau en gérant le déplacement coordonné de l'ensemble objet-robots tout en communiquant *via* la base la plus proche (BASE), la position de l'objet à un opérateur distant elle-même connectée à un réseau filaire (LAN ou WAN). Il recevra en retour la consigne de déplacement de l'ensemble pour l'instant suivant. Les robots communicants doivent transmettre les informations capteurs au robot maître en respectant des contraintes temps réel liées à la dynamique des lois de commande. Aussi est-il important d'étudier l'influence d'un retard variable sur la commande d'un sous-système en termes de stabilité.

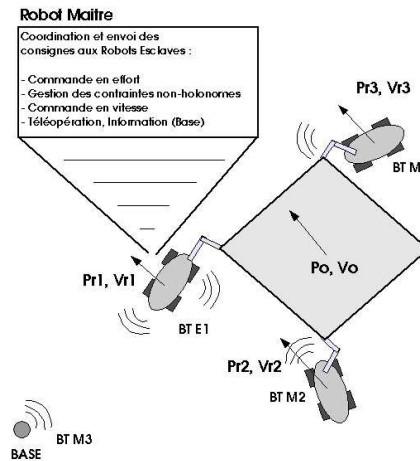


Figure 2. Commande coordonnée multi-robots

2.2. Stabilité d'un sous-système du 1^{er} ordre avec un retard constant

L'ensemble des manipulateurs mobiles est constitué d'actionneurs électriques que l'on peut modéliser par un modèle linéaire continu dont les non-linéarités sont représentées par des perturbations extérieures. Partant de cette hypothèse, il est ainsi possible d'écrire un modèle linéaire continu constitué par un ensemble de sous-systèmes indépendants linéaires continus. Dans le cadre de cette étude et pour des raisons de

simplicité du calcul de la stabilité, nous opterons pour un modèle équivalent du premier ordre pour la modélisation des actionneurs électriques. C'est une approximation grossière que nous justifierons pleinement à la section suivante grâce à l'utilisation d'une loi de commande par mode glissant.

Considérons donc, un ensemble de sous-systèmes linéaires indépendants du 1^{er} ordre de la forme :

$$\tau_i \dot{q}_i(t) + q_i(t) = k_{vi}(u_i(t) - d_i(t)) \quad [1]$$

Avec q_i la variable de sortie du sous-système i (roues, direction, articulation du manipulateur..), k_{vi} le gain statique entre l'entrée électrique du variateur et la position de sortie de l'actionneur, $u_i(t)$ la tension de commande, τ_i la constante de temps équivalente du sous-système i et $d_i(t)$ une perturbation extérieure. Cet ensemble de sous-systèmes linéaires indépendants constitue le système distant. Il peut être commandé suivant deux modes différents.

Le premier est un mode classique (cf. figure 3) où le réseau de transmission est inclus dans une boucle de commande de haut niveau (ex : boucle de commande dans l'espace cartésien, commande coordonnée avec d'autres robots mobiles).

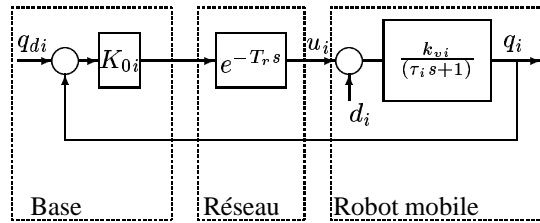


Figure 3. *Modèle de transmission*

Le second, est un mode de téléopération où la boucle distante est fermée par l'opérateur. Ce deuxième mode est moins contraint en ce qui concerne les conditions de stabilité. Aussi considérons-nous les résultats de l'analyse de stabilité du premier mode afin d'établir une relation entre le retard et la stabilité du système. Nous prenons comme hypothèse le retard, aller-retour T_r présent dans la transmission des données, constant. On obtient ainsi l'expression suivante :

$$T_r \leq \frac{\tau_i \arccos(-\frac{1}{K_{0i}k_{vi}})}{K_{0i}k_{vi} \sin \arccos(-\frac{1}{K_{0i}k_{vi}})} \quad [2]$$

Avec K_{0i} le gain proportionnel du sous-système i de la boucle de commande externe. Si nous faisons l'hypothèse que ces gains K_{0i} ainsi que k_{vi} sont constants, on arrive au résultat suivant :

$$\frac{T_r}{\tau_i} \leq C_i \quad [3]$$

où C_i est une constante positive non nulle. Ce résultat peut nous amener à limiter la bande passante du sous-système distant *via* une boucle de commande locale lorsque le retard moyen T_r augmente. Ce résultat sur la stabilité de la commande d'un système du premier ordre est intéressant. En effet, la limite de stabilité est linéaire en fonction du retard. Cependant, est-il possible que le comportement dynamique d'un moteur à courant continu puisse être assimilé à un système du premier ordre ? Est-il possible de plus d'ajuster la bande passante de la boucle fermée distante ? Dans le cas où la boucle de commande locale sur le système distant permet de forcer le comportement dynamique, cette hypothèse sera justifiée. Une méthode de commande robuste permet de forcer la dynamique du système commandé sur une surface de glissement. Il s'agit de la commande par régime glissant.

2.3. Adaptation de la bande passante du sous-système distant en fonction du retard en utilisant une loi de commande par mode glissant d'ordre supérieur

2.3.1. Mode glissant d'ordre supérieur

Une solution pour forcer dynamiquement la réponse d'un système pour qu'il réagisse comme un premier ordre avec une bande passante prédéfinie est d'utiliser une loi de commande par mode glissant [Utkin, 1977]. Cette technique contraint le système à suivre le comportement d'une surface de glissement $s = 0$. Cependant, le mode de glissement est obtenu à partir d'un vecteur de commande discontinu qui a pour conséquence d'introduire un phénomène de chattering.

Beaucoup de travaux ont été menés dans ce domaine pour trouver des solutions sans vraiment de résultats satisfaisants sauf pour la méthode des modes glissants d'ordre supérieur [Fridman, Levant, 2002]. En effet, cette méthode préconise d'annuler la surface de glissement ainsi que ses dérivées d'ordre supérieur jusqu'à l'ordre $(r - 1)$ (cf. équation 4), r étant le degré relatif du système (cf. équation 5).

$$s = \dot{s} = \ddot{s} = \dots = s^{r-1} = 0 \quad [4]$$

Le degré relatif est défini par l'équation suivante :

$$\frac{\partial s^i}{\partial u} = 0, \frac{\partial s^r}{\partial u} \neq 0 \quad [5]$$

L'équation 5 permet de déterminer l'ordre relatif du système commandé et de connaître à quel niveau la discontinuité va intervenir. Dans le cas, par exemple, d'un système d'ordre relatif $r = 1$, pour éviter le phénomène de chattering une commande glissement d'ordre 2 pourra être utilisée et ainsi c'est la dérivée par rapport au temps $\dot{u}(t)$ qui sera considérée comme le nouveau vecteur de commande. Ainsi le vecteur de commande $u(t)$ sera alors continu et le phénomène d'oscillation limite autour du point d'équilibre aura disparu.

2.3.2. Stratégie de commande

Considérons la représentation d'état d'un moteur électrique à courant continu :

$$\begin{cases} \dot{x}_{1i} = x_{2i} \\ \dot{x}_{2i} = -\frac{1}{\tau_i}x_{2i} + \frac{1}{\tau_i}k_{vi}u_i(t) \end{cases} \quad [6]$$

où $x_{1i} = q_i$ (avec $i = 1..6$) est la position articulaire du i^{me} actionneur, τ_i est le temps de réponse du moteur et $u_i(t)$ la variable de commande. Nous proposons une surface du premier ordre avec la valeur de la sortie désirée $x_{1di} = 0$:

$$s_i(x_i) = -x_{2i} - \lambda_i x_{1i} \quad [7]$$

La commande en mode glissant d'ordre 2 considère la dérivée temporelle de la variable de commande $\dot{u}(t)$ comme le vecteur de commande. Une solution est obtenue en exprimant \ddot{s}_i avec l'équation (6). La seconde équation est obtenue en considérant la dérivée temporelle du modèle d'état telle que la dérivée de la variable de commande $\dot{u}_i(t)$ apparaît à l'intérieur de \ddot{s}_i . Dans le second cas, nous obtenons, le modèle suivant :

$$\begin{cases} \dot{x}_{1i} = x_{2i} \\ \dot{x}_{2i} = x_{3i} \\ \dot{x}_{3i} = -\frac{1}{\tau_i}x_{3i} + \frac{1}{\tau_i}k_{vi}\dot{u}_i(t) \end{cases} \quad [8]$$

En étudiant ce nouveau modèle décrit par l'équation 8, nous proposons une nouvelle surface de glissement s_{yi} dépendant de la fonction $s_i(x_i, t)$ telle que :

$$s_{yi} = y_{2i} + \alpha_i y_{1i} \quad [9]$$

où $y_{2i} = \dot{s}_i$ and $y_{1i} = s_i$. Cette fonction peut être exprimée par :

$$s_{yi} = -x_{3i} - (\alpha_i + \lambda_i)x_{2i} - \alpha_i \lambda_i x_{1i} \quad [10]$$

La dérivée du vecteur de commande $\dot{s}_i(t)$ peut être décrite comme la somme de deux fonctions :

$$\dot{u}_i(t) = \dot{u}_{eqi} + \Delta \dot{u}_i \quad [11]$$

où \dot{u}_{eqi} est la commande équivalente pour obtenir $\dot{s}_i = 0$:

$$\dot{u}_{eqi} = \frac{\tau_i}{k_{vi}} \left(\frac{x_{3i}}{\tau_i} - (\lambda_i + \alpha_i)x_{3i} - \alpha_i \lambda_i x_{2i} \right) \quad [12]$$

et $\Delta \dot{u}_i$ est le second terme discontinu habituellement utilisé pour compenser les dynamiques non modélisées :

$$\Delta \dot{u}_i = K_i \operatorname{sgn}(s_{yi}) \quad [13]$$

où K_i est la valeur d'entrée maximale des actionneurs. L'équation d'atteignabilité η (14) est nécessaire pour avoir une convergence en temps fini.

$$s_{yi} \dot{s}_{yi} \leq -\eta |s_{yi}| \quad [14]$$

Les équations (12) et (13) vérifient la condition (14) qui assure la convergence en temps fini. En intégrant l'équation (14), nous obtenons le résultat suivant :

$$t_{ei} \leq \frac{\alpha_i \lambda_i \tau_i}{k_{vi} K_i} \quad [15]$$

où t_e est la limite supérieure de la convergence en temps fini telle que $s_i(t_{ei}) = 0$. En utilisant les équations (11), (12) et (13), le vecteur global de commande $u_i(t)$ peut être exprimé par :

$$u_i(t) = u_{eqi}(t) + \int K_i \operatorname{sgn}(s_{yi}) dt$$

Ce vecteur de commande est continu et on peut donc supposer que le phénomène de *chattering* est éliminé.

2.3.3. Stabilité

Choisissons la fonction de Lyapunov suivante :

$$V(\mathbf{X}, t) = \frac{1}{2} \mathbf{X}^t \mathbf{P} \mathbf{X}$$

avec $\mathbf{X} = [x_1, x_2, x_3]^T$ le vecteur d'état et \mathbf{P} une matrice définie par :

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & -\alpha_i \lambda_i \\ 0 & \alpha_i \lambda_i - \frac{\alpha_i \lambda_i}{\tau_i} & -(\alpha_i + \lambda_i) \\ -\alpha_i \lambda_i & -(\alpha_i + \lambda_i) & -1 \end{bmatrix}$$

La dérivée de cette fonction peut être exprimée par :

$$\dot{V} = (-\alpha_i - \lambda_i + \frac{1}{\tau_i}) x_{2i}^2 + \frac{\alpha_i \lambda_i}{\tau_i} x_{1i} x_{3i} - \frac{k_{vi} K_i}{\tau_i} |s_{yi}|$$

où V est définie positive et \dot{V} est définie négative sous les conditions suivantes :

$$\frac{\alpha_i \lambda_i}{\alpha_i + \lambda_i} < \frac{1}{\tau_i} \quad [16]$$

Nous pouvons donc conclure que si les conditions de stabilité de l'équation 16 sont respectées, le système est exponentiellement stable.

2.3.4. Résultats de simulation

Les hypothèses et les calculs présents ont été validés en simulation en utilisant un modèle Simulink d'un moteur électrique à courant continu (cf. équation 6) commandé par un algorithme d'ordre 2. Nous avons testé ce modèle en réglant les coefficients λ_i et α_i afin d'atteindre la limite de stabilité. Les résultats sont présentés à la figure 4. Ces résultats sont obtenus pour $\tau_1 = 0,1s$, $K_1 = 15$ et $k_{v1} = 1rd/V$. Nous avons comparé cette courbe à la courbe de stabilité théorique. On peut effectivement conclure que cette condition est validée.

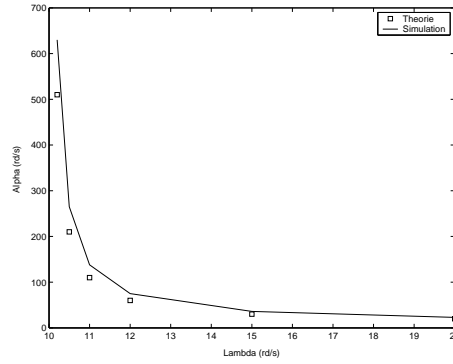


Figure 4. *Limite de stabilité*

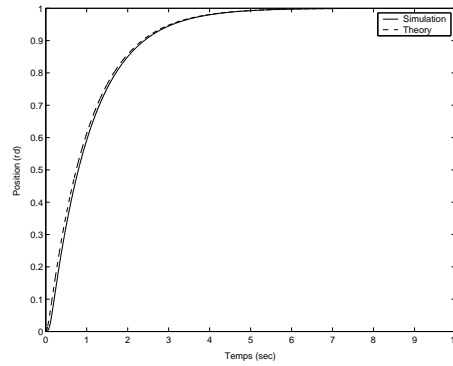


Figure 5. *Réponse à l'échelon*

La réponse temporelle à un échelon de position est présentée à la figure 5. Elle est comparée à la réponse temporelle théorique obtenue lorsque la surface de glissement $s_{y1} = 0$. Lorsque cette condition est vérifiée on obtient la réponse temporelle suivante $x_{1T}(t)$:

$$x_{1T}(t) = u(t) \left(1 - \frac{\alpha_1}{\alpha_1 - \lambda_1} e^{-\lambda_1 t} + \frac{\lambda_1}{\alpha_1 - \lambda_1} e^{-\alpha_1 t} \right)$$

Dans le cas où $\lambda \gg \alpha$, on obtient une réponse qui peut être approximée à celle d'un premier ordre :

$$x_{1T}(t) \approx u(t) (1 - e^{-\alpha_1 t})$$

Ce qui permet alors de vérifier l'équation 3, avec dans ce cas, $\alpha_i = \frac{1}{\tau_i}$. La figure 6 présente dans le domaine temporel et dans l'espace de phase le comportement du système soumis à un échelon lorsque les coefficients de la surface de glissement sont modifiés pendant le mouvement ($\alpha_1(1) = 0, 4rd/s \rightarrow \alpha_1(2) = 2rd/s$ et $\lambda_1 = 20rd/s$).

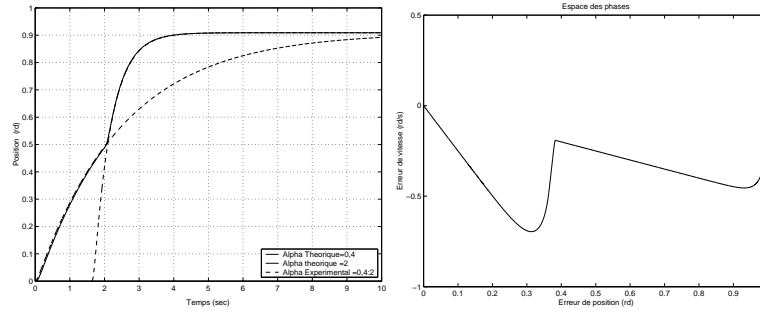


Figure 6. *Commutation de surface*

La figure 7 représente un test de l'ensemble de la boucle lorsque le système distant reçoit un ensemble d'ordres de l'opérateur par une suite d'échelons lorsque brusquement à $t=12s$, le retard moyen T_r change et le régulateur mode glissant s'adapte en conséquence.

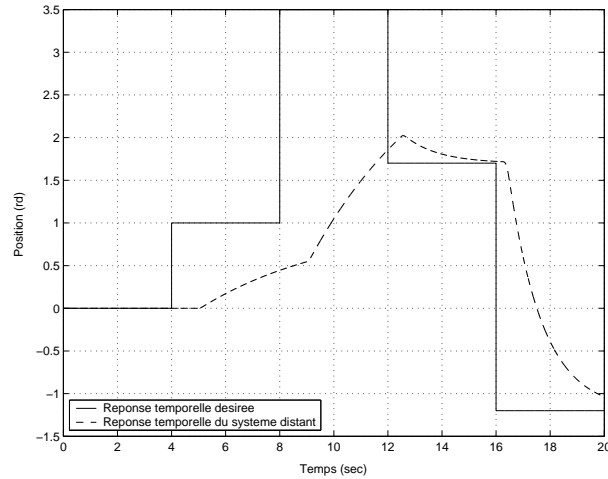


Figure 7. *Adaptation de la bande passante en fonction du retard moyen T_r*

Le retard moyen change brusquement de la valeur $T_r(1) = 1s$ à $T_r(2) = 0,4s$. Le coefficient α passe automatiquement de la valeur $\alpha(1) = 0,4$ à $\alpha(2) = 2$ et ce, afin de vérifier la relation de stabilité définie précédemment. Cette approche de la commande en présence de retard variable et les caractéristiques de Bluetooth nous permettent d'envisager l'utilisation de cette technologie sans fil dans le cadre de la commande à distance de robots mobiles.

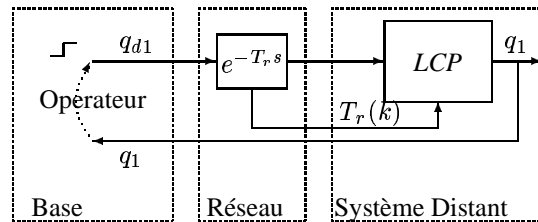


Figure 8. *Tâche de Téléopération*

En effet, la méthode que nous venons d'exposer peut se résumer par la figure 8 où la loi de commande en position (LCP) locale, basée sur la commande par régime glissant adapte le coefficient $\alpha(k)$ en fonction du retard $T_r(k)$ afin d'obtenir le produit $T_r(k) \cdot \alpha(k) = Cste$, avec k étant l'instant discret (kT_t) d'apparition des échantillons. Cette constante dans le cas de la simulation avait été prise à $C = 0,2$. Cette méthode permet même de s'affranchir des problèmes de coupure de la communication où les retards de transmission peuvent prendre des valeurs très importantes. Dans ce cas précis la valeur de α sera très faible et tendra vers zéro. Le système aura un temps de réponse qui tendra vers l'infini. On passe ainsi de manière continue du fonctionnement nominal à l'arrêt du véhicule en fonction de la qualité du lien de communication.

3. Bluetooth et robotique mobile, quelques cas d'exploitation

Dans le cadre de la robotique mobile, le robot embarque un dispositif de capteurs lui permettant d'avoir des informations tant proprioceptives qu'extéroceptives. Les odomètres et gyromètres permettent par exemple au robot de calculer sa position de manière relative (avec l'inconvénient de cumuler l'erreur). Les capteurs d'efforts lui donnent également une information relative, sur son interaction avec l'environnement (paroi ou objet quelconque). Les systèmes tels le GPS (*Global Positioning System*), les télémètres lasers ou ultrasons, etc, contribuent à un positionnement absolu (*i.e.* par rapport à un repère connu). Les capteurs visuels (caméras) fournissent également des informations sur la localisation du robot lui-même par rapport à son environnement, ou des objets de l'environnement par rapport au robot, mais ils induisent des contraintes particulières de par, entre autres, la quantité de données générées et la complexité du traitement requis pour en extraire les informations pertinentes (en comparaison avec des capteurs moins « sophistiqués »). Le robot, un manipulateur mobile par exemple, comprend également un ensemble d'actionneurs, principalement des moteurs, pour la locomotion (porteur) et la manipulation (bras). Ces dispositifs, capteurs et actionneurs, sont respectivement rapportés à des entrées et sorties d'un contrôleur embarqué assimilable au « cerveau » du robot (organe contrôlant ses mouvements et son comportement). L'échange de signaux ou d'informations entre ces entités embarquées est usuellement basé sur des liaisons filaires, sérielles ou parallèles, selon des topolo-

gies différentes telles les liaisons point à point ou bus. Les progrès technologiques sur les liaisons sans fil, de type WLAN (*Wireless Local Area Network*) comme WiFi par exemple (802.11b), ou de type WPAN (*Wireless Personal Area Network*) comme Bluetooth, permettent d'envisager une mutation architecturale. Sur ce plan architectural, la topologie (bus, point à point, étoile, etc.), le débit, les plages de fréquences, etc. ne sauraient être considérés comme critères de choix sans porter attention au profil du réseau (nature des échanges supportés et modèle de coopération). Dans le cadre d'un système de commande, le profil déterministe du réseau, notamment au niveau de l'accès au médium et du modèle de coopération sous-jacent, est indispensable à l'obtention de garanties sur les contraintes spatio temporelles des échanges entre les entités constituant l'architecture globale. En effet, les relations entre les entités embarquées sur le robot sont :

- périodiques (voire synchrones) à l'image par exemple d'une boucle d'asservissement impliquant la chaîne capteurs-contrôleur-actionneurs (cf. section 2) ;
- aperiodiques (voire événementielles) dès lors qu'on intègre des entités dites « intelligentes » capables de communiquer de leur propre initiative un phénomène (événement) détecté localement (un détecteur de proximité repérant et signalant la présence d'un obstacle par exemple).

L'utilisation dans ce contexte de la technologie Bluetooth, présentée plus en détails dans la suite de cet article, pourrait permettre la construction d'architectures plus évolutives, tout en répondant aux besoins énoncés, de par ses canaux synchrones et asynchrones et son modèle de coopération maître-esclaves. Néanmoins, nous allons plutôt nous attacher à travers cette communication, à l'utilisation de cette technologie pour supporter les échanges entre le robot et son monde extérieur ; ceci comprenant les interactions entre robots et celles entre le robot et son environnement. Ces différents cas reposent sur l'exploitation de différentes topologies accessibles *via* Bluetooth (ces topologies sont détaillées section 3.1).

3.1. Le système de communication sans fil Bluetooth

3.1.1. Présentation générale du WPAN

Bluetooth est un système de transmission de données sans fil à courte portée par liaison radio. Ce système fait partie de ce que l'on appelle maintenant les WPAN pour *Wireless Personal Area Network* [802.15, 2002]. En effet, et malgré les tentatives d'évolution récentes, il ne faut pas assimiler Bluetooth à un WLAN (*Wireless Local Area Network*), car il n'a pas été étudié pour assurer les services offerts par Wifi, Wifi5 et autre Hiperlan2, mais plutôt pour évoluer conjointement avec ces réseaux locaux sans fil [Jacquet *et al.*, 2000]. Le but initial de Bluetooth est bien de remplacer les fils et les câbles autour des équipements électroniques communicants, et non pas de remplacer le réseau Ethernet sur lequel est connecté l'ordinateur multimédia. Les débits, les portées, les protocoles, les topologies et le nombre de nœuds associés, n'ont pas été prévus pour réaliser un véritable réseau local. C'est initiale-

ment la société Ericsson qui a initié le projet Bluetooth, dans le but avoué d'offrir aux combinés GSM un moyen de communication sans fil vers des périphériques tels que des kits mains libres ou des PDA. Aujourd'hui, Ericsson s'est associé au sein d'un groupement, le SIG (Special Interest Group of Bluetooth), avec les principaux leaders des télécommunications et de l'informatique tels que IBM, Intel, Lucent, Microsoft, Motorola, Nokia ou Toshiba... Ce groupement a déjà proposé plusieurs versions de la spécification Bluetooth, accessibles gratuitement sur le site web officiel [SIG, 2002]. Actuellement, la spécification dans sa version v1.2 est divisée en deux documents :

- les spécifications principales des couches de protocoles et caractéristiques physiques et matérielles (Core part specification v1.2),
- les protocoles et procédures nécessaires pour les différentes applications (Profiles part specification v1.2).

3.1.2. *Caractéristiques physiques*

A l'instar du WLAN 802.11 dont il est en partie issu, Bluetooth utilise la bande radio ISM (Industriel, Scientifique, Medical) à 2.4 GHz. Le choix des concepteurs de Bluetooth s'est fait entre les trois couches physiques radio qui sont présentes dans la norme 802.11 :

- une couche IR basée comme son nom l'indique sur le médium infrarouge [Val *et al.*, 2003]. Cette couche n'a jamais donné le jour à des produits finis,
- une couche radio basée sur l'étalement de spectre par séquençement direct (DSSS),
- et une couche radio basée sur l'étalement de spectre par sauts de fréquences (FHSS).

C'est cette dernière qui a servi de base de travail aux concepteurs de Bluetooth. La bande totale s'étend de 2.402 GHz à 2.480 GHz, et est divisée en 79 canaux de 1MHz. Avant 2001, la France était limitée à 23 canaux pour éviter des interférences avec les bandes radio de l'armée, qui est en train de les libérer progressivement. Trois classes de puissance ont été retenues :

- la classe 3 pour les très courtes portées (~ 10 cm) avec une puissance maximum de sortie de 0dBm,
- la classe 2 pour les courtes portées (~ 10 m) avec une puissance maximum de sortie de 4dBm. C'est cette classe qui sera la plus couramment utilisée, par exemple entre un GSM et un kit oreillette main-libre,
- la classe 1 pour les portées plus importantes (~ 100 m) avec une puissance maximum de sortie de 20dBm. Cette classe est prévue pour assurer à Bluetooth une portée comparable aux WLAN existants (802.11b, 802.11a ou 802.11g). Ces classes 1 et 2 pourront plus particulièrement être employées dans la communication entre robots mobiles.

Une modulation GFSK (Gaussian Frequency Shift Keying) assure un débit binaire instantané nominal de 1 Mbps. Remarquons qu'il ne faut surtout pas confondre ce

débit nominal avec les débits utilisables qui, comme nous le verrons par la suite, sont bien en deçà. Parmi les 79 canaux disponibles, une séquence de sauts de 32 canaux est choisie par l'initiateur de la communication (appelé maître ou Master et dont nous préciserons le rôle par la suite). Toutes les $625 \mu s$, l'émetteur et le(s) récepteur(s) changent de fréquence porteuse simultanément pour passer au slot suivant dans la séquence de 32 (cf. figure 9).

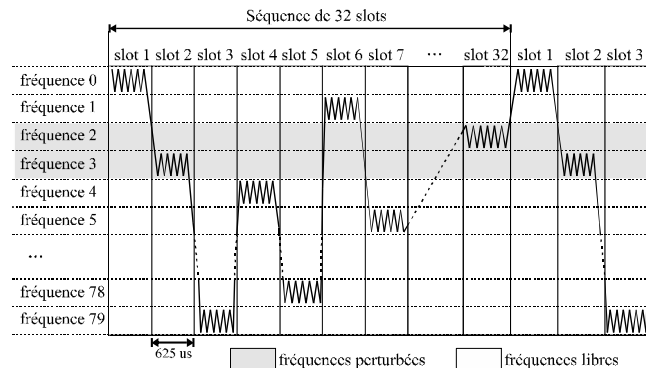


Figure 9. Principe du saut de fréquences de Bluetooth

Ce principe du FHSS assure ainsi une bonne immunité aux bruits associés aux plages de fréquences perturbées à un instant donné (les 32 canaux sont choisis non contigus), ainsi qu'un premier niveau de protection contre le piratage. D'autre part, le FHSS permet le fonctionnement simultané et à proximité de plusieurs maîtres avec des séquences différentes, en limitant au maximum les perturbations mutuelles. Pour ce faire, chaque maître choisit une séquence de sauts qui lui est propre car dépendant entre autre de son adresse physique unique (appelé BDAddress dans la spécification Bluetooth, d'une longueur de 48 bits). Les séquences de sauts de chaque maître ont la particularité de ne disposer que de quelques fréquences communes entre elles, ce qui évite des collisions potentiellement longues entre maîtres voisins. La dernière version v1.2 de Bluetooth prévoit même d'éviter les fréquences perturbées (par exemple Wifi) en retirant de la séquence de 32 les précédents slots parasités. L'étalement de spectre est alors appelé AFH pour Adaptive Frequency Hopping. Associé à ce principe de modulation physique par étalement de spectre par sauts de fréquences, la couche supérieure nommée Baseband utilise une transmission TDD (Time Division Duplex) qui permet une communication bi-directionnelle alternative entre 2 nœuds Bluetooth (le maître et un autre nœud). L'initiateur de la communication utilise le premier slot pour parler vers son interlocuteur qui lui répond au slot suivant (cf. figure 10). Dans le cas le plus simple, chaque paquet de données utilise qu'un slot, mais il est possible pour certains types d'échanges de regrouper des slots par 3 ou 5 pour disposer de paquets plus longs. La figure 10 illustre des échanges entre un maître et 2 interlocuteurs qualifiés tout naturellement d'esclaves (ou slave).

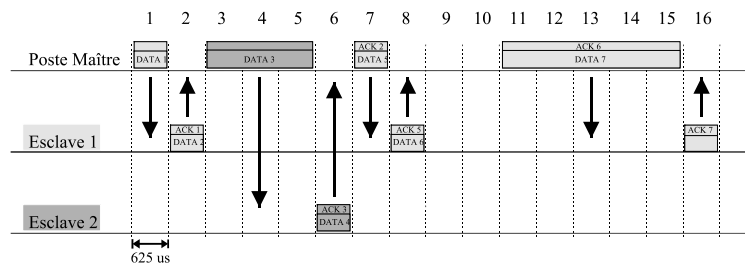


Figure 10. Principe du TDD

On peut remarquer que grâce à ce regroupement de slots par 1, 3 ou 5, le maître émet les données toujours à partir des slots impairs, l'esclave lui, utilise toujours les slots pairs.

3.1.3. La topologie maître-esclave

Toute communication Bluetooth est donc nécessairement basée sur la notion de maître-esclave. Ceci impose des contraintes d'architectures sur les différentes topologies du réseau Bluetooth. La première et la plus simple des topologies proposées par Bluetooth est une liaison point à point entre un maître et un esclave (cf. figure 11).

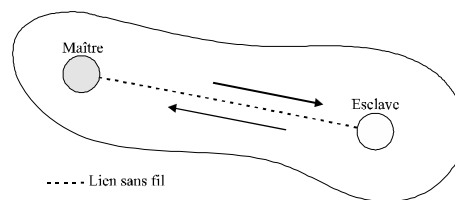


Figure 11. Liaison point à point

Le maître est l'initiateur de la communication. C'est lui qui synchronise son esclave à partir de la séquence de sauts qu'il a choisie en fonction de son adresse propre. Différents échanges d'informations sont nécessaires avant la phase de transfert de données proprement dite. Ces informations initiales servent à détecter la présence de l'esclave ainsi qu'à échanger des paramètres sur les horloges respectives des 2 interlocuteurs. Ceci correspond aux deux phases nommées Inquiry et Paging. C'est également à ce niveau que l'on peut ajouter des techniques d'authentification et de cryptage pour assurer un certain niveau de sécurité dans les échanges sans fil [Pinel, 2001]. Ce type de communication point à point est à l'heure actuelle celui qui est implémenté dans tous les premiers produits Bluetooth, par exemple entre 2 cartes PCMCIA Bluetooth

pour Notebooks, 2 dongles USB Bluetooth ou encore entre un GSM et un kit oreillette [Ericsson, 2002].

3.1.4. Les autres topologies Piconet et Scatternet

Dans la topologie Piconet, le maître a la possibilité de gérer des communications avec plusieurs esclaves situés dans sa zone de couverture radio (cf. figure 12). Cette topologie permet donc la mise en réseau de 8 nœuds Bluetooth (au maximum) : 1 maître et 7 esclaves. Tous les esclaves sont synchronisés par le maître en suivant la même séquence de sauts. Aucune communication n'est possible directement entre deux esclaves. C'est au maître à gérer les relais de trames au niveau supérieur entre deux esclaves qui désirent s'échanger des données. De la même façon, le maître doit également assurer, en fonction des besoins de l'application, les interrogations successives des esclaves. Différentes stratégies peuvent être employées [Lee, Gerla, 2001] : tour de rôle, notion de priorité.

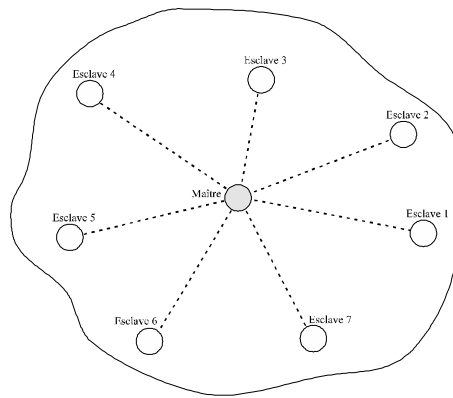


Figure 12. *Le piconet*

Face aux critiques des fabricants et utilisateurs de WLAN, les concepteurs de Bluetooth ont prévu dans la spécification une troisième topologie. Cette dernière permet d'obtenir théoriquement un nombre plus élevé de nœuds, pouvant, sous diverses conditions, communiquer entre eux [Haartsen, 2000]. Cette topologie, appelée Scatternet (cf. figure 13), est en fait l'union de 10 Piconets au maximum. Cette union permet de porter de 8 à 80 le nombre de nœuds. Chaque Piconet est toujours géré par un maître unique, avec 7 esclaves actifs au maximum. Pour assurer des échanges entre nœuds de Piconet différents, il est nécessaire de disposer de stations communes entre ces derniers. Deux configurations sont possibles :

- soit le nœud commun est alternativement esclave de 2 maîtres de 2 Piconets différents (esclave B-1/esclave A-3). Ceci implique une "désassociation" de l'esclave avec son ancien maître (maître A), puis une association avec le nouveau maître (maître

B). Des trames spécifiques sont prévues pour assurer ces fonctions mais la décision sera laissée à la charge des couches supérieures,

– soit le nœud commun est le maître d'un Piconet et l'esclave de l'autre Piconet voisin (maître C/esclave B-3). Avant de pouvoir lui-même devenir esclave de l'autre maître du Piconet voisin (maître B) en acceptant sa synchronisation basée sur une autre séquence de sauts, le maître doit alors obligatoirement déléguer temporairement son rôle de maître à un de ses esclaves (esclave C/maître C). Sinon, le Piconet privé de maître ne peut plus fonctionner.

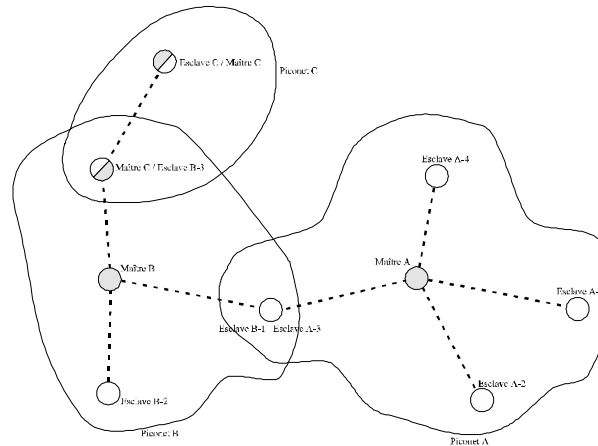


Figure 13. *Scatternet*

Dans les deux cas, d'importants temps de commutation sont à prévoir, ralentissant d'autant les échanges entre nœuds fortement distants. Dans ces conditions, des protocoles de relais de trames sont à prévoir au niveau supérieur. La mise en œuvre d'un routage dynamique en fonction de la topologie plus ou moins complexe du Scatternet paraît, de plus, absolument nécessaire car par définition, le Scatternet est basé sur des nœuds sans fil pouvant être mobiles. Cette topologie n'est pour l'instant, que très rarement implémentée dans les premiers produits Bluetooth. Elle devrait faire l'objet d'importants travaux de recherche et développement, en particulier pour des applications de robotiques mobiles qui disposeraient de plus de 8 éléments mobiles communicants. La nouvelle version v1.2 de Bluetooth envisage d'élargir cette notion de Scatternet en permettant à chaque nœud d'être maître et esclave tout au long des échanges d'informations.

3.1.5. *Les liens data asynchrones*

Deux grands types d'applications potentielles ont conduit les initiateurs du projet Bluetooth à prévoir dès le début deux catégories de liens, ceux principalement dédiés au transport de la voix numérisée, nommés SCO (*Serial Connection Oriented*), et ceux initialement prévus pour le transport de données des applications informatiques, nom-

més ACL (*Asynchronous Connection Less*). Les utilisations typiques de cette dernière catégorie peuvent être :

- la liaison entre 2 ordinateurs portables lors de transferts de fichiers,
- la synchronisation entre un ordinateur de poche et un ordinateur de bureau,
- l'impression de documents sur une imprimante Bluetooth,
- les échanges de données de type contrôle-commande au sein même d'une machine,
- la liaison sans fil entre un ordinateur mobile et un modem jouant le rôle de point d'accès internet.

Dans tous ces cas, des volumes importants de données sont échangés. Les besoins en débit sont typiquement de 50 à 700 kbps, parfois dissymétriques, comme cela peut être le cas pour un accès « http » entre un client mobile et un serveur fixe distant. Pour ces liaisons de données, la notion de la qualité de service est non primordiale. Ces liens sont donc asynchrones et sans connexion.

3.1.6. *Les liens voice synchrones*

Pour les liaisons de type voix, un débit de 64 kbps symétrique assure une qualité d'écoute équivalente au standard ISDN (*Integrated Services Digital Network*) filaire. La qualité de service associée à cette catégorie requiert entre autre, une bande passante garantie, ainsi que l'absence de gigue dans les retards de propagation et d'accès au médium. Ces fortes contraintes temporelles sont satisfaites par le caractère synchrone de ces liaisons associées à un mode orienté connexion. Cette seconde catégorie de liens est nommée SCO pour *Serial Connection Oriented*. La période T_{sco} d'accès au canal est respectivement de 2, 4 et 6 fois $625 \mu s$ pour les liens HV1, HV2 et HV3 (*High quality Voice 1-2-3 slots*). Ainsi, ces trois liens HVi offrent le même débit utile de 64 kbps, bien qu'ils n'offrent pas la même longueur de champ de données utiles, à cause de la différence de redondance induite par des codes correcteurs différents. Les signaux analogiques vocaux sont numérisés au moyen d'un code PCM-CVSD (*Pulse Code Modulation-Continuous Variable Slope Delta Modulation* [Fraisie *et al.*, 1999]) avec un échantillonnage sur 1 bit à 64 KHz.

3.2. *Choix de Bluetooth face aux autres solutions sans fil*

Après avoir présenté rapidement les caractéristiques prometteuses de ce nouveau système de communication sans fil Bluetooth, il convient néanmoins de justifier notre choix face aux nombreuses autres solutions de communications sans fil qui existent aujourd'hui.

3.2.1. *Modems radio point à point*

Bien que nos premiers travaux portent sur une topologie point à point sans fil entre un organe de pilotage et un seul robot mobile, nous envisageons par la suite de propo-

ser des solutions réseaux plus conséquentes, en permettant des communications sans fil avec une flotte complète de robots et/ou avec des éléments actifs de l'environnement. Les solutions point à point de modems sans fil, travaillant pour la plupart dans la bande ISM des 433 Mhz, sont donc à proscrire, puisque qu'elles ne sont pas équipées pour la plupart d'une couche MAC permettant un accès au médium contrôlé évitant les collisions de trames radio entre les nombreuses entités communicantes concurrentes. La solution consistant à rajouter une méthode d'accès est guère envisageable, face aux nombreuses solutions beaucoup plus performantes, validées et normalisées. Ces dernières sont bien évidemment celles regroupées autour de la norme IEEE 802.11, et qualifiées à tort d'Ethernet sans fil.

3.2.2. *WLAN 802.11*

Le groupe de travail 802.11 a depuis 1996 développé et normalisé toute une suite de protocoles de communications appartenant aux réseaux locaux sans fil WLAN. L'impossibilité avec du sans fil de détecter une collision par écoute de la porteuse et comparaison avec son propre signal émis (aveuglement) a entraîné la naissance d'une nouvelle méthode d'accès CSMA/CA, utilisée par toutes les normes issues de 802.11. Les premiers produits offraient des débits de 1 et 2 Mbps en saut de fréquence ou en séquençement direct. C'est l'avènement de la norme 802.11b qui a démocratisé ces WLAN, sous l'appellation commerciale WiFi, en offrant 11Mbps au maximum. Ces solutions travaillent dans la bande ISM 2.4 Ghz. La bande 5Ghz est utilisée principalement aux EU (WiFi5) et offre des débits jusqu'à 54 Mbps, au prix d'une portée moindre. De nouveaux produits correspondant à la norme 802.11g offrent ces mêmes débits (voire même 2*54Mbps/s) tout en restant compatibles avec WiFi puisque travaillant dans la même bande. La quasi-totalité de ces produits est dédiée à une utilisation orientée bureautique, avec des cartes PCMCIA, PCI, MMC, USB ou Ethernet... Il est très difficile de concevoir une carte de taille réduite, de très faible consommation, alors qu'il est plus facile de le faire avec Bluetooth. De plus, pour l'instant, aucune qualité de service ne peut être associée aux liens 802.11, puisque seul le mode DCF a été implémenté. Il existe bien un mode PCF dans la norme, qui est théoriquement prévu pour assurer une QoS en termes de débit et de temps d'accès au médium radio, mais aucun driver ne permet son exploitation. Enfin, l'utilisation de produits 802.11 rend quasi obligatoire la présence d'un driver de communication implémentant les couches basses logicielles de WiFi. Ces drivers sont pour l'instant disponibles pour environnement Windows et Linux. L'utilisation d'un microcontrôleur est alors bien plus délicate, et ne peut se faire qu'au prix d'un développement spécifique de driver. L'accès aux couches basses câblées, gage de rapidité et de souplesse, pour l'implémentation d'un protocole aussi spécifique que ceux que nous désirons faire, est ainsi quasi impossible.

Toutes ces constatations ont orienté notre choix sur le WPAN Bluetooth, amplement suffisant pour les débits et les portées désirés ; les caractéristiques d'un WLAN 801.11 n'étant à nos yeux, non nécessaires face aux contraintes de poids, de taille et de consommation imposées par les robots de nos applications.

En ce qui concerne les applications les plus gourmandes en débit, on retrouve bien évidemment les transferts temps réel vidéo. Avant de poursuivre nos investigations sur le WPAN Bluetooth, nous avons prouvé la possibilité de transférer sur un lien BT un flux vidéo en streaming offrant une qualité vidéo amplement suffisante pour un retour image lors d'un pilotage à distance d'un robot mobile équipé d'une caméra.

3.3. Interactions entre robots reposant sur une topologie point à point

A titre d'illustration, considérons par exemple deux robots manipulateurs mobiles impliqués dans l'accomplissement d'une tâche en coopération, en l'occurrence une manipulation d'objet à deux bras (figure 14). La commande des manipulateurs mobiles pour l'accomplissement d'une telle tâche implique des variables d'état et des mesures (capteurs) issues des deux robots et génère des commandes ou des consignes (selon l'architecture retenue) à destination de ces manipulateurs mobiles. Ce type de commande, *i.e.* d'asservissement, basé sur l'échange de « données » entre les robots, suppose le respect des contraintes temporelles qui conditionnent la validité de la commande dont sa stabilité. Disposer d'une liaison synchrone, à disponibilité et bande passante garanties, serait idéal. L'exploitation des canaux synchrones de Bluetooth pour le transport de données autres que la voix est encore à l'étude. Par conséquent, l'utilisation des canaux asynchrones pour ce type de commande est une solution de repli qui requiert une caractérisation précise de la qualité de service offerte (cf. section 4.2.4). Même si la qualité de service n'est pas garantie (retards, pertes), il reste possible dans une certaine mesure, d'assurer la stabilité de la boucle de commande (cf. section 2.2).

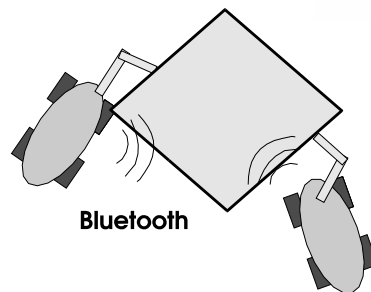


Figure 14. Boucle de commande impliquant Bluetooth

Dans l'exemple représenté sur la figure 14, un robot a le rôle de maître et l'autre celui d'esclave, et ce tant sur l'aspect modèle de coopération (maître/esclave au sens Bluetooth, cf. section 3.1.3) que sur l'aspect plus fonctionnel. Prenons l'exemple suivant : les robots coopèrent dans le transport d'une charge. Le robot dit maître impose

le déplacement et le robot dit esclave suit ce déplacement en maintenant un effort nul dans la direction du déplacement (entre lui et la charge).

3.3.1. *Interactions entre le robot et son environnement reposant sur une topologie Piconet*

Supposons un robot mobile évoluant dans un environnement d'intérieur, pour la surveillance d'un bâtiment par exemple. La navigation dans ce type d'environnement structuré s'appuie souvent sur une cartographie du lieu (un plan) au sein duquel le robot est amené à se repérer. Embarquer le plan de tout un site est difficilement envisageable sur de petits robots mobiles pour des raisons justement de capacité restreinte d'un système embarqué. Cela revient à dédier le robot à un site, avec la nécessité d'actualiser ces données embarquées à chaque modification structurelle du lieu.

L'autre apport potentiel d'une technologie Bluetooth se situe donc dans l'échange d'informations entre le robot se déplaçant sur le site et des « balises » disposées sur le site, sources de ces informations. La figure 15 représente un robot mobile pénétrant dans une pièce où est disposée une balise. En tant que maître (maître/esclave au sens Bluetooth), il enrôle l'esclave qu'il rencontre au sein d'un Piconet qu'il construit donc dynamiquement. Il peut alors procéder à l'échange d'informations relatives à cette pièce (parois, issues, etc.). Notons que nous illustrons ce cas *via* l'échange de plans mais il est extrapolable à l'échange, bidirectionnel cette fois-ci, entre un opérateur distant et le robot. Cela nous amène vers une architecture réseau à plusieurs niveaux (cf. section 5) au sein de laquelle les balises sont reliées entre elles par un réseau filaire tel Ethernet par exemple ou un réseau sans fil à plus longue portée (WLAN) comme 802.11.

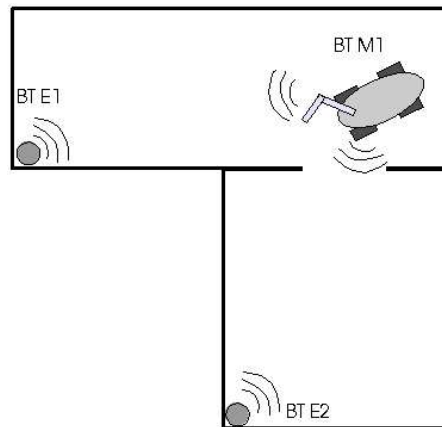


Figure 15. *Acquisition d'informations sur l'environnement textitvia Bluetooth*

3.4. Interactions entre des robots reposant sur une topologie Scatternet

En poursuivant les illustrations précédentes, il est possible d'envisager la coopération de plusieurs robots mobiles au sein d'un même site. Un robot était jusqu'ici considéré comme maître. Nous sommes alors en présence de plusieurs maîtres, capables d'alterner entre les rôles de maître et d'esclave dans le cadre de la topologie Scatternet de Bluetooth (cf. section 3.1.4). Ils peuvent donc échanger des informations aussi bien entre eux qu'avec leur environnement (cf. figure 16).

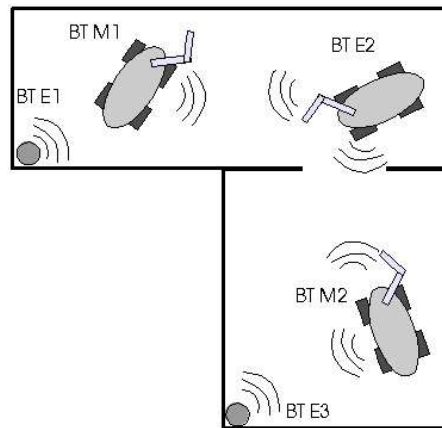


Figure 16. Bluetooth comme support des interactions entre robots

Si différentes illustrations exposées ici restent prospectives, notamment au regard de l'évolution de la norme Bluetooth (le Scatternet entre autres), nous avons néanmoins progressé sur l'exploitation de Bluetooth comme médium pour la coopération « directe » de robots (*i.e.* en liaison point à point). Nous allons maintenant présenter la particularité de la commande à distance *via* un réseau sans fil afin de pouvoir juger ultérieurement de l'adéquation de Bluetooth avec ce contexte.

4. Etude, réalisation et mesures

L'utilisation de Bluetooth pour assurer des échanges sans fil d'informations entre un robot mobile et un poste de pilotage a nécessité tout d'abord la réalisation d'une plate-forme matérielle puis logicielle de test et de prototypage de liens Bluetooth. Ces études nous ont ainsi permis d'identifier l'adéquation Bluetooth-Robotique.

4.1. Réalisation d'une plate-forme Bluetooth matérielle

Il est encore aujourd'hui très difficile de s'approvisionner en composants Bluetooth. De plus, les quelques modèles disponibles sont essentiellement conçus pour une utilisation exclusive avec des ordinateurs de type PC, soit disposant d'une interface PCMCIA, soit USB. Les systèmes d'exploitation sont alors Windows 98/ME/2K/XP ainsi que les dernières versions de Linux. Ces cartes Bluetooth ne sont absolument pas ouvertes et il est impossible de s'en servir pour une implémentation sur système industriel comme un robot dans notre cas. Notre travail initial a donc été de concevoir une plate-forme matérielle de développement, nous permettant également une phase de métrologie afin d'identifier les canaux Bluetooth les plus aptes à assurer le transport sans fil de données pour notre application de contrôle-commande temps réel. Nous avons conçu un système modulaire, présenté dans sa globalité figure 17, offrant différents types d'interfaces possibles au niveau HCI (*Host Controller Interface*) entre les couches basses et les couches logicielles de la pile protocolaire Bluetooth. Le cœur du système est une carte fille (#1) intégrant les couches physique radio, LMP (*Link Manager Protocol*) et Baseband (bande de base). Cette carte peut recevoir tout module Bluetooth Ericsson (ROK) interchangeable [Ericsson, 2002] via un support à force d'insertion nulle. Une antenne intégrée est prévue. L'ensemble des broches du ROK est disponible sur un connecteur vers la carte interface. Suivant les versions de ROK, certaines broches ne sont pas connectées en interne car associées à des fonctions non implémentées (par exemple le port USB pour les ROK 101008). Nous avons pris soin de déporter toutes les broches possibles en faisant l'union de celles disponibles dans les différentes versions de module Bluetooth, et ceci tout en assurant une compatibilité de brochage.

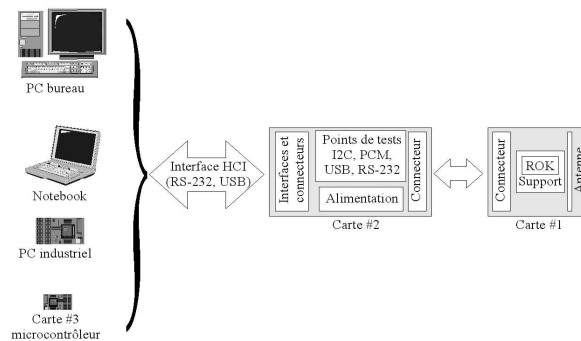


Figure 17. Conception modulaire de cartes

Une seconde carte (#2) offre l'ensemble des interfaces HCI de la spécification Bluetooth. Des composants assurent les conversions et adaptations de tensions. Cette carte permet également un accès facilité à tous les signaux électriques par des points de tests (USB, RS-232, PCM). Enfin, c'est cette carte qui assure l'alimentation, soit par l'intermédiaire d'une alimentation externe qui est régulée et stabilisée par des composants de la carte, soit directement par le port USB. Suivant les versions, divers connecteurs normalisés permettent la connexion des interfaces HCI (par exemple un DB9 pour les port RS-232). Une troisième carte (#3) intègre les couches hautes logicielles et l'application elle-même. Plusieurs microcontrôleurs sont actuellement à l'étude, dont le PIC 16F877 de Microchips. Dans une version ultérieure encore plus compacte, il est prévu d'intégrer sur la même carte les modules #2 et #3, voire même les 3 cartes #1, #2 et #3. D'autre part, pour des raisons d'ergonomie de développement évidentes, nous avons également utilisé à la place de la carte #3 un ordinateur de bureau, un notebook et un ordinateur de type PC industriel d'Advantech, le PCM-5820 [Advantech, 2002].

4.2. Conception d'une plate-forme Bluetooth logicielle

La conception d'une plate-forme matérielle a été suivie tout naturellement d'une phase plus informatique puisque correspondant à la conception d'une plate-forme Bluetooth logicielle. La pile protocolaire Bluetooth propose la présence d'une couche logicielle nommée L2CAP (*Logical Link Control and Adaptation Protocol*) au dessus de l'interface HCI. Cette couche de niveau 2 assure le rôle classique de couche 2 du modèle OSI (*Open System for Interconnection*) mais aussi l'adaptation de protocoles vers des couches supérieures standards telles que PPP, TCP/IP...

4.2.1. Niveau L2CAP

Différentes piles logicielles sont compatibles avec le matériel que nous venons de présenter [Val *et al.*, 2003]. Nous avons choisi d'utiliser la pile logicielle CSTACK [Val, Juanolle, 2002]. Le logiciel, libre de droit pour une utilisation universitaire, est disponible sous la forme de composants Active-X pour environnement Microsoft Windows, utilisables à partir d'applications développées avec Visual Studio (C++, Visual Basic). La pile CSTACK est compatible avec tout composant Bluetooth certifié, en particulier les modules ROK 101007 et 101008. L'interface HCI retenue et implémentée est le port série à un débit de 57600bps. Initialement développée comme composant Active-X offrant des API au-dessus de la couche L2CAP, la pile CSTACK est également disponible maintenant au niveau inférieur HCI. Le composant Active-X proposé dans cette première version permet donc le développement d'applications respectant l'architecture protocolaire présentée figure 18. Le composant offre un ensemble complet de primitives d'accès aux fonctions L2CAP qui peuvent être réparties en 4 classes :

- les primitives de gestion de la pile,
- les primitives de gestion de la connexion,

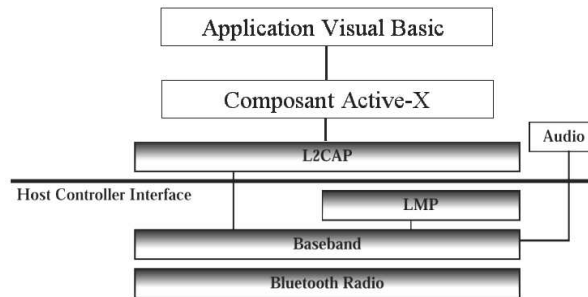


Figure 18. Architecture protocolaire associée au composant Active-X L2CAP

- les primitives de transfert de données,
- les primitives de gestion des événements.

4.2.2. Mesures temporelles haut niveau

Nous avons développé plusieurs applications permettant de mettre en œuvre des liens Bluetooth point à point. Nous présentons par la suite, une des applications de test et de métrologie réalisée. Cette dernière permet de rechercher un module distant, d'ouvrir des canaux L2CAP, d'émettre et de recevoir des données et de mesurer divers paramètres afin d'évaluer les performances de la plate-forme. L'application développée permet également une mesure des principaux délais et débits. Une librairie spécifique [Kainka, Berndt, 1999] nous a permis de mesurer les temps avec une précision de l'ordre de la microseconde. Le tableau 1 représente un exemple typique des résultats obtenus grâce aux applications développées. Les tests ont été effectués sur 2 ordinateurs PC Pentium III 800 Mhz et Pentium I 200 Mhz sous Windows 98 SE en alternant le rôle de maître et d'esclave, afin d'évaluer également l'incidence possible du temps de traitement de l'application et de la couche L2CAP logicielle. Chaque PC est relié par un port série à 57600 bps à une des cartes Bluetooth présentée à la section 4.1. Nous avons fait varier la distance entre les cartes de 1 à 10 m, sans détecter une influence notable pour les missions de classe 2.

Cette première série de mesures permet les commentaires et analyses suivants. La recherche de l'adresse locale est rapide. Elle est fonction de l'ordinateur qui effectue cette recherche. Nous mesurons donc essentiellement le temps de traversée de la couche logicielle L2CAP locale. La recherche de l'adresse distante est majoritairement fonction du temps nécessaire à la découverte du nombre de modules déportés. Ces 10 secondes sont imposées par les spécifications Bluetooth. Le temps nécessaire à l'obtention des adresses correspondantes et le choix d'un module parmi ceux découverts (ici 1 seul) est alors négligeable. Le temps d'ouverture du premier canal L2CAP est très long par rapport à une réouverture ultérieure ou à l'ouverture d'autres canaux avec le même module esclave. La couche L2CAP gère donc une seule fois les pro-

blèmes de synchronisation d'horloge entre le maître et l'esclave. Ceci est vérifié si aucune modification du nombre de modules déportés du maître ne se produit. Ces modifications éventuelles sont détectables par l'indication DevsListChange. Les débits mesurés semblent peu fonction de la rapidité du processeur du maître. Il est fort probable que le temps de traversée de la couche L2CAP et de l'interface HCI soit très pénalisant. L'implémentation du composant Active-X nous impose un débit limité à 57600 bps sur l'interface HCI RS232 proposée. Ceci représente alors un fort goulot d'étranglement. De plus, la couche L2CAP ne nous offre pas le choix du type de paquet (DM1, DM3, présenté à la section 2.4.4), car elle nous impose l'utilisation d'un paquet DM1 offrant un débit théorique de 108 Kbps. Ce dernier point n'est ici pas critique puisque, de toute façon, on ne peut pas espérer obtenir plus que le débit de l'interface HCI à 57600 bps, ce qui ne sera pas le cas lorsque l'on pourra monter le débit du port série à la limite maximum de 460 Kbps pour les nouveaux ROK bientôt disponibles, ou lorsque la pile CSTACK offrira l'utilisation de l'interface USB à 2 Mbps.

Machine de mesure maître	Pentium III	Pentium I
Temps de recherche @ locale	10500 μs	27040 μs
Temps de recherche @ distante	10346000 μs	10370000 μs
Temps d'ouverture 1 ^{er} canal	2849656 μs	2295226 μs
Temps d'ouverture 2 ^{eme} canal	326182 μs	326254 μs
Temps de réouverture de canal	326134 μs	326532 μs
Temps de fermeture 1 ^{er} canal	363669 μs	342841 μs
Temps de fermeture 2 ^{eme} canal	377383 μs	355224 μs
débit L2CAP de bout en bout	34622 bps	34720 bps

Tableau 1. Résultats des mesures temporelles L2CAP sous Windows 98

En conclusion, sur la couche L2CAP proposée par CSTACK et sur les applications que nous avons développées, nous pouvons identifier plusieurs avantages :

- la plate-forme logicielle proposée est d'une utilisation très simple et permet le développement rapide d'applications de mesures et de mise en œuvre de liens Bluetooth ;
- nous nous sommes limités à des liens point à point asynchrones uniquement à cause des limitations des versions de ROK à notre disposition, mais la plate-forme logicielle est compatible avec la notion de Piconet en synchrone et asynchrone ;
- le développement d'interfaces homme-machine pour des applications de communication Bluetooth est possible pour tout équipement informatique utilisant les systèmes d'exploitation Windows 98/ME/2K/XP. Le passage vers WinCE et LINUX devrait être tout aussi aisé ;

Par contre, il apparaît plus intéressant, ou tout du moins très complémentaire, de pouvoir disposer d'une pile protocolaire Bluetooth permettant de nous affranchir des retards induits par la couche L2CAP, de la lenteur générée par un système d'explo-

tation lourd couplé d'une application gourmande en temps d'exécution et en capacité mémoire. Cette dernière remarque (entre autres) nous a conduit à développer une pile Bluetooth dédiée aux microcontrôleurs embarqués que nous présentons au paragraphe suivant.

4.2.3. Niveau HCI

L'idée directrice du travail présenté dans ce paragraphe est de disposer de nœuds Bluetooth de très petites tailles, autonomes, non tributaires de la présence d'une unité centrale de type PC de bureau, Notebook ou même PC industriel. Un microcontrôleur semble tout indiqué pour assurer l'implémentation des couches logicielles de Bluetooth et de l'application. L'accès aux couches les plus basses possibles dans la pile protocolaire doit permettre un contrôle et un choix plus adaptés des liens de bas niveau, ainsi qu'une garantie d'optimisation des temps de traversée des couches. Dans une première étape, nous avons voulu valider pour notre plate-forme, la possibilité d'implémenter une pile minimale disposant de possibilités de mesures temporelles en temps réel. Par souci initial de facilité de conception, nous avons développé la pile pour processeur Intel 8x86, en prenant soin de la rendre facilement adaptable à tout type de microcontrôleur, ce qui a aussi été réalisé par la suite pour un microcontrôleur PIC. Le protocole de niveau HCI est assez différent et plus complexe que celui présenté précédemment concernant la couche L2CAP. Afin de pouvoir appréhender son fonctionnement avant toute implémentation logicielle, nous avons développé et utilisé différents outils et méthodes. L'utilisation du logiciel de tests et de monitoring BSK fourni avec le Starter Kit d'Ericsson a permis une première approche intuitive. Ce logiciel offre la possibilité de piloter en ligne les modules Bluetooth, de visualiser et mémoriser les retours lors des différentes phases du protocole, mais également de créer des scripts pour automatiser les échanges. Nous avons développé, pour appréhender et étudier ces commandes, un analyseur Bluetooth permettant un « espionnage » en temps réel, c'est-à-dire l'affichage et l'analyse des trames HCI lors d'échanges Bluetooth. Cet analyseur vient se connecter en dérivation entre le module ROK esclave et l'ordinateur hôte maître, sur le lien RS-232. Il récupère les niveaux électriques des 2 signaux TxD et RxD. Nous développons actuellement une version pour port USB, ainsi que le module logiciel de décodage, d'interprétation et d'affichage, en fonction des différentes couches de protocole Baseband, LMP ou L2CAP.

4.2.4. Mesures temporelles bas niveau

Afin de permettre le développement rapide d'applications de tests et de mise en œuvre de liens LMP, nous avons conçu un macrolangage de programmation assembleur offrant plusieurs macro-instructions que l'on peut répartir en 2 classes :

- les primitives de gestion du protocole,
- les primitives de mesures de performances temporelles.

Voici à titre d'exemple le genre d'analyse que l'on peut réaliser à partir d'applications développées à l'aide de ce langage. La figure 19 représente de manière graphique la durée totale Emission+Réception vers un module ROK de la commande RbdAddr

en fonction du débit du port série. Cette commande HCI permet la lecture de l'adresse locale du ROK. Le temps d'exécution de cette commande avec le retour de l'adresse est directement fonction du débit du port série mais également du temps de réponse du ROK qui doit rester plus ou moins constant. Cette courbe exponentielle laisse penser que l'on ne gagnera pas beaucoup en temps d'exécution de la commande en augmentant le débit au delà de 230,4 Kbps. Ces mesures seront néanmoins complétées par la suite avec la nouvelle version de ROK à 460,8 Kbps (voire même 921,6 Kbps).

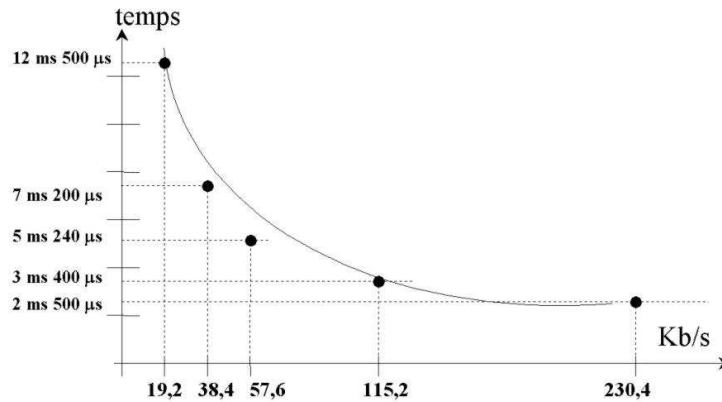


Figure 19. *Mesure temporelle de la commande "RbdAddr"*

Afin de réaliser une première comparaison entre les applications L2CAP développées sous Windows avec la pile CSTACK, et les applications développées sous DOS avec notre pile assembleur au niveau HCI, nous avons réalisé, lorsque cela était possible, les mêmes mesures que celles présentées dans le tableau 1, en conservant le même débit de 57600 bps au niveau de l'interface HCI. Le tableau 2 résume l'ensemble des mesures réalisées au niveau HCI, et rappelle les mesures obtenues au niveau L2CAP pour les commandes équivalentes. On peut noter la grande reproductibilité des mesures obtenues. Ayant remarqué lors des mesures précédentes la faible influence de la rapidité du processeur des PC connectés aux ROK maître et esclaves, ces mesures se sont déroulées sur deux machines homogènes équipées d'un processeur Pentium III 550 Mhz.

La recherche de l'adresse locale est deux fois plus rapide qu'au niveau L2CAP. Ici, elle se limite à l'émission et réception d'une trame. Le débit du port série a une influence importante comme nous l'avons remarqué précédemment, mais un temps constant minimal est à prévoir pour laisser le temps au ROK local de répondre. La recherche de l'adresse distante est toujours proche de 10s, même si on semble gagner en moyenne 0,1s par rapport au niveau L2CAP. La création d'une connexion asynchrone (DMi ou DHi pour $i=1, 3$ ou 5) entre le maître et l'esclave est environ deux fois plus rapide que l'ouverture du premier canal L2CAP. Il est, au niveau HCI, possible de créer

Niveau protocolaire	HCI (ASM)	L2CAP (CSTACK)
Temps de recherche @ locale	5,24 ms	10,5 ms
Temps de recherche @ distante	10,2 s	10,34 s
Temps de création d'une connexion ACL	1,192 s	2,85 s
Temps de création d'une connexion SCO	468 ms	*
Temps de déconnexion ACL	268 ms	363ms
Temps de déconnexion SCO	266 ms	*

Tableau 2. *Résultats des mesures temporelles sous DOS avec la pile assembleur HCI*

des connexions synchrones HVi (choix impossible au niveau L2CAP *), en associant un lien SCO à un lien ACL déjà créé. Les horloges entre le maître et l'esclave étant désynchronisées, cette création de lien synchrone (pour l'instant uniquement vocal) ne prend que 468 ms en moyenne. Les déconnexions ACL et SCO sont d'une durée très similaire, en moyenne 267 ms, donc plus rapides que la fermeture d'un canal L2CAP.

Nous avons également réalisé différentes mesures concernant le débit utile offert par les différents canaux asynchrones. Nous avons eu la possibilité de faire varier, grâce à des primitives assembleurs HCI, le débit du port série entre le PC et le ROK. Malheureusement, à cause de la version P13A des ROK à notre disposition à l'heure actuelle, il n'est pas encore possible de disposer de débits certifiés supérieurs à 115200bps, même si dans certains cas expérimentaux, le débit de 230 Kbps a été atteint. Ce débit HCI de 115200bps ne permet pas d'éliminer le goulot d'étranglement précédemment évoqué sauf peut-être pour un lien DM1 limité théoriquement à 108 Kbps. On obtient pourtant pour les applications HCI, un débit Bluetooth utile de bout en bout supérieur à celui constaté au niveau L2CAP. Par exemple, pour 57600 bps sur le port série, le débit utile est de 40 Kbps ; pour 115200 bps sur le port série, on obtient alors un débit utile de bout en bout entre le PC maître et le PC esclave de 79 Kbps. Ces valeurs sont quasiment les mêmes quel que soit le type de lien asynchrone utilisé (DMi et DHi), ce qui prouve bien que l'on mesure plus le débit HCI que le débit de la couche physique radio.

Notre volonté d'intégrer un lien Bluetooth dans une boucle fermée d'asservissement à distance d'un robot mobile sans fil nous a conduit à mesurer et étudier le temps d'aller et retour de trames entre deux modules Bluetooth. Afin d'obtenir les temps les plus faibles, nous avons principalement développé des applications de test en utilisant les primitives HCI de la pile assembleur, même si quelques mesures ont été effectuées avec la pile CSTACK pour comparaison. Dans un premier temps donc, nous avons centré nos mesures sur l'utilisation des liens DM1. Plusieurs applications ont été réalisées où nous mesurons le délai entre le moment où un ordinateur A émet une trame de données vers un ordinateur B distant, et le moment où la réponse de B est parvenue à l'ordinateur A de départ. Nous avons fait varier le débit du port série entre chaque ordinateur et module ROK, en testant les valeurs 38400, 57600 et 115200 bps. Par extrapolation, nous avons estimé les délais correspondants aux débits 230400 et 460800

bps. Dans chaque cas, trois longueurs différentes de trames ont été programmées : 1, 50 et 100 octets de données utiles dans chaque trame aller et retour. La figure 20 représente graphiquement la synthèse de ces mesures. Il est intéressant de noter que pour des trames portant 50 octets utiles, on obtient des délais aller-retour voisin de 40ms, ce qui, pour notre application, semble être compatible et utilisable. L'influence du débit est bien sûr très faible dans le cas de trames réduites à un seul octet utile. Ce dernier cas ne présente d'ailleurs que peu d'intérêt applicatif mais a été évalué avec l'objectif de connaître la borne inférieure du délai. Les mesures faites pour des trames de 100 octets mettent en évidence les limites supérieures sur la longueur des trames. En effet, ces dernières entraînent des délais pouvant être jugés trop grands, surtout pour des débits sur le port série ne dépassant pas 57600 bps. En ce qui concerne les mesures réalisées avec la pile CSTACK, elles sont comme prévu bien moins rapides. Nous obtenons pour les tailles de données utiles de 1, 50 et 100 octets, des délais respectifs de 101, 149 et 179 ms. Dans ces trois cas, une augmentation du délai par rapport aux mesures à 57600bps effectuées avec la pile assembleur au niveau HCI de 60 à 5 ms est sans doute liée au temps de traversée de la couche L2CAP implémentée dans la pile CSTACK. En conclusion sur ces mesures, il semble donc qu'il soit préférable d'utiliser le niveau HCI pour espérer obtenir les délais les plus courts, gage de rapidité et de réaction de la boucle d'asservissement souhaitée.

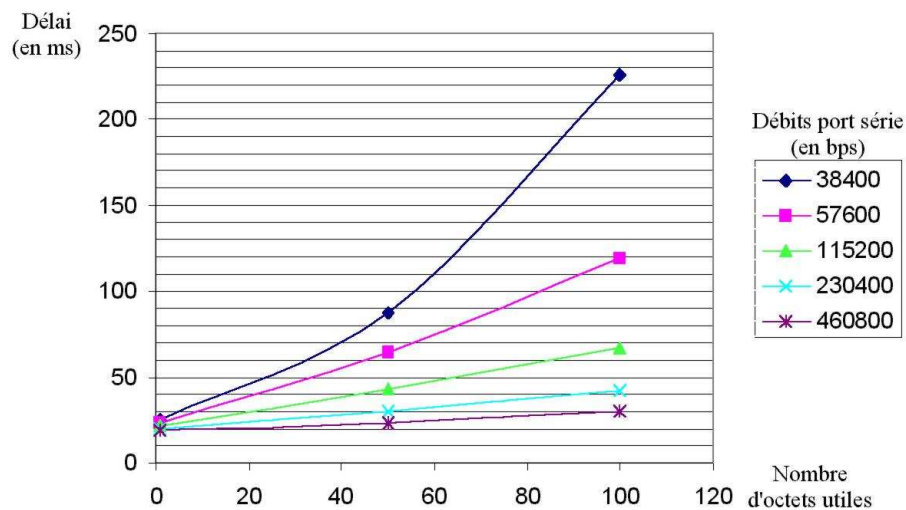


Figure 20. *Mesure du délai aller-retour*

4.2.5. Mesures de portées

Nous avons été amenés à réaliser plusieurs campagnes de mesures pour évaluer la portée potentielle de notre plate-forme Bluetooth, et l'incidence de la distance sur

la qualité du lien. Bien que pouvant utiliser des amplificateurs de classe 1 permettant d'atteindre les 100m de portée, nous avons limité nos tests à des émetteurs de classe 2 classiques. Les 10m de portée en champ libre sont atteints sans problème, associés à un taux d'erreur bit BER de 10^{-6} . Au-delà de cette distance, la liaison se dégrade très rapidement et coupe en quelques dizaines de centimètres supplémentaires. Ceci est facilement explicable par l'utilisation des codes correcteurs d'erreurs FEC (*Forward Error Correction*) intégrés aux liens ACL employés. Tant que le taux d'erreur reste dans la limite acceptable de l'efficacité du code correcteur, le lien reste robuste, au-delà de 10m, tout s'effondre. De la même façon, nous n'avons pas pu mesurer de réels écarts de taux d'erreur en faisant varier la distance entre émetteur et récepteur dans la zone de couverture inférieure à 10m d'un émetteur Bluetooth. Ceci est un gage de qualité et surtout d'homogénéité des liens dans leur domaine de travail nominal.

4.3. Identification de l'adéquation Bluetooth - Robotique

Ce dernier paragraphe consacré à notre contribution proprement dite est, en fait, une analyse des résultats obtenus en termes de métrologie et mesures lors de la caractérisation des liens Bluetooth, face aux besoins applicatifs de contrôle-commande de robot. Notre but est d'identifier la meilleure adéquation possible entre les caractéristiques judicieuses de Bluetooth (type de lien, topologie, niveau d'interface...) face au cahier des charges induit par l'aspect robotique.

4.3.1. Choix de la topologie adéquate

Nous nous sommes tout d'abord placés dans le cas initial de la présence d'un seul robot mobile associé à un unique organe fixe de pilotage. C'est donc tout naturellement la topologie point à point de Bluetooth qui semble la plus appropriée. A partir de cette dernière, il est alors nécessaire de choisir qui sera le maître, et qui sera l'esclave. Même si ce choix n'est pas capital pour l'aspect communication, nous pouvons néanmoins proposer une orientation. Le poste de pilotage fixe est sans doute celui qui disposera de la plus forte capacité informatique, par exemple dans un PC sous Windows ou Linux. Le robot, lui, disposera d'une intelligence plus limitée, par exemple implémentée dans un microcontrôleur spécialisé. Les tâches du maître étant plus nombreuses, en particulier lors de l'initialisation de la connexion Bluetooth, il est préférable qu'elles soient exécutées par le poste fixe. Le robot se trouve alors dévolu au rôle d'esclave Bluetooth.

4.3.2. Choix de la couche protocolaire Bluetooth

La couche protocolaire au-dessus de laquelle l'application de contrôle-commande doit se situer est également un paramètre important. Bluetooth n'a pas été prévu initialement pour ce genre d'application. Notre contribution doit également proposer un choix. La couche L2CAP est principalement utilisée pour des applications nécessitant le transport d'une quantité importante de données utiles, comme cela peut être le cas pour un transfert de fichier. Elle offre de plus une plus grande facilité de pro-

grammation en limitant à une trentaine le nombre de primitives d'accès. Ceci se fait néanmoins au prix d'un ralentissement et d'un temps de traversée non négligeable pour des applications industrielles à forte contraintes temporelles où le temps de propagation est critique. De plus, L2CAP ne permet pas le choix entre les différents liens offerts par Bluetooth. Seuls des liens DM1 peuvent être utilisés, ce qui semble bien restrictif, en particulier lorsque l'on pourra utiliser les liens synchrones. Notre choix s'est donc porté sur le niveau HCI qui présente plusieurs avantages. Il est assez facile de concevoir une pile protocolaire compatible avec ce niveau. Nous ne sommes alors absolument pas contraints par la disponibilité limitée des piles protocolaires pour processeur spécialisé. Il est de plus possible de choisir en fonction du type de message, le lien Bluetooth adapté (nous allons y revenir ci-dessous). Enfin, c'est à ce niveau le plus bas dans la partie logicielle de la pile, qui offre l'alternative du niveau d'interfaçage, que l'on peut espérer obtenir des temps de traversée les plus faibles, même en présence des microcontrôleurs à capacité mémoire et temps d'exécution limités.

4.3.3. Choix du type de liens Bluetooth

Au niveau HCI, on a la possibilité de pouvoir choisir pour l'instant entre différents liens ACL. Ces derniers offrent en point à point, la même qualité de service en termes de disponibilité du canal, puisque seuls le maître et son esclave l'utilisent. Le choix reste donc à faire entre des paquets DMi et DHi. Tout dépend alors de la qualité de service que l'on désire en termes de rapidité d'accès au canal, en termes de quantité d'information et de fiabilité du lien. Le tableau 3 fait la synthèse des liens ACL disponibles pour l'instant.

Nom	Type	données	Code détec.	Code cor.	Taille	débit max.
DM1	ACL	18 octets	CRC	2/3 FEC	1 slot	108 Kbps
DH1	ACL	28 octets	CRC	aucun	1 slot	173 Kbps
DM3	ACL	123 octets	CRC	2/3 FEC	3 slots	387 Kbps *
DH3	ACL	185 octets	CRC	aucun	3 slots	586 Kbps *
DM5	ACL	226 octets	CRC	2/3 FEC	5 slots	478 Kbps *
DH5	ACL	341 octets	CRC	aucun	5 slots	723 Kbps *

Tableau 3. Les différents liens ACL

– Les paquets DM offrent un meilleur taux d'erreur résiduel sans pour autant faire appel systématiquement à une réémission du mode ARQ puisque le code 2/3 FEC permet la correction automatique d'une erreur simple pour 10 bits utiles transmis. Ceci se fait au prix d'un plus faible débit utile par rapport aux paquets DH.

– Il peut être tentant de vouloir, sans réfléchir, utiliser des paquets de taille plus importante (3 ou 5 slots). Ceci va permettre d'obtenir un débit plus important. Dans le cas d'application de contrôle-commande, ce paramètre n'est peut-être pas si capital que cela, les quantités d'informations restant faibles.

– De plus, les paquets n'utilisant qu'un seul slot (Dx1) offrent une périodicité d'utilisation du canal plus faible, donc un retour d'information plus rapide. Dans le

cas d'un asservissement en boucle fermée, ce paramètre peut être primordial pour la stabilité.

– Bien que cela ne soit pas noté sur le tableau 3, il est possible d'utiliser des liens ACL non symétriques. Par exemple, un paquet DH5 dans le sens fixe-mobile pour la commande, et un paquet DH1 dans le sens mobile-fixe pour le retour. Les débits représentés dans la colonne de droite et marqués par une * sont ceux qui correspondent à des liens symétriques.

4.3.4. *Choix de l'interface HCI*

Actuellement, peu de composants discrets sont disponibles sur le marché. Notre choix s'est porté sur la famille de composants Ericsson, seuls disponibles il y a 3 ans. D'autres sont maintenant apparus, ou devraient l'être. Pourtant, Ericsson est encore l'un des rares à offrir un ensemble complet d'outils de développement de cartes Bluetooth. Le composant principal est le ROK. Ce dernier présente encore plusieurs lacunes. L'interface RS-232 est limitée à des débits certifiés de 115200 bps, ce qui est un goulot d'étranglement important comme nous l'avons mesuré. L'interface USB est opérationnelle bien que non certifiée pour les ROK 101008. Cette interface doit offrir des débits de 12 Mbps et corrige les lacunes identifiées sur l'interface série. Pourtant, seuls les drivers pour Windows 98/ME/2K/XP sont disponibles. Ceci va en totale contradiction avec la volonté d'utiliser des microcontrôleurs, en particulier pour l'intelligence embarquée dans un robot. Il semble très délicat, voire impossible, de réaliser soi-même un driver USB spécifique, sans disposer de renseignements techniques de la part d'Ericsson. L'interface I^2C n'est pas, à ce jour, opérationnelle.

5. Perspectives

Nous allons terminer cet article en évoquant diverses perspectives de recherches sur lesquelles nous nous orientons dès à présent, et qui ont été rendues possibles par les travaux présentés dans ce papier. La maîtrise d'une plate-forme mixte (logicielle et matérielle) de développement et de tests du WPAN Bluetooth nous a permis un apprentissage rapide et pratique de ce nouveau système de communication radio. Nous nous sommes forcés de le rendre le plus ouvert possible afin de pouvoir réaliser les mises à jour nécessaires et attendues, en particulier sur l'évolution des ROK, de leur interface HCI, de leur firmwares...

5.1. *Prise en compte de plusieurs robots : Piconet et Scatternet*

La disponibilité récente des ROK 101007 va nous permettre de prendre en compte la topologie Piconet et Scatternet. Ceci aura une retombée directe sur la possibilité de pilotage de plusieurs robots esclaves au moyen d'un maître (Piconet) ou de plusieurs (Scatternet). Il reste alors à proposer, simuler et implémenter des protocoles applicatifs temps réel permettant une répartition adéquate des ressources de communication en

fonction des demandes liées aux pilotages en boucle fermée de plusieurs robots, et aux interactions qu'il peut y avoir entre ces derniers.

5.2. Architecture réseau à plusieurs niveaux

D'autres travaux sont en cours sur la proposition d'une architecture de communication à plusieurs niveaux. Ceci devrait nous permettre de prendre en compte des épines dorsales locales basées sur des liens filaires (Ethernet ou bus de terrain). Nous envisageons par la suite de prolonger vers internet *via* des routeurs, les liens de bout en bout. Enfin, pour se soustraire à la portée et au débit limités sur les liens sans fil offerts par Bluetooth, il est également possible d'utiliser en complément Bluetooth, le WLAN IEEE 802.11b. Dans les dernières versions de Wifi, 802.11a offre 54Mbps dans la bande radio 5 GHz, ce qui limiterait les incompatibilités électromagnétiques possibles avec Bluetooth. 802.11g+ offre même un débit double de 108Mbps, ce qui est amplement suffisant pour nos applications industrielles actuelles, mais gage d'évolution possible vers des besoins futurs multimédias.

5.3. Utilisation des liens synchrones à QoS

Bien que les canaux synchrones aient été initialement prévus pour le transport de la voix, il peut paraître très intéressant de les utiliser pour véhiculer les données à forte contrainte temporelles. Ceci ne sera possible que lorsque les modules ROK offriront la possibilité d'émission et réception de données *via* l'interface HCI. Pour l'instant, seule la couche audio accède aux liens SCO *via* un CODEC. Dans notre cas initial de lien point-à-point entre le poste de pilotage fixe et le robot, il n'y a pas de problème lié à l'occupation du canal ACL (le seul disponible pour l'instant au niveau de l'interface HCI). Lorsque nous serons en présence de plusieurs robots, ceci ne sera plus vrai. Il faudra alors gérer une méthode d'accès déterministe pour l'utilisation du canal de communication par les différents nœuds. Une solution possible sera alors d'utiliser les liens SCO qui offrent de façon native cette QoS.

6. Conclusion

La prise en compte du réseau de transmission sans fil Bluetooth dans la coordination de manipulateurs mobiles pour une tâche complexe telle que le transport d'objet lourd et encombrant, est une voie intéressante et prometteuse à explorer. En effet, l'augmentation des débits et de la qualité de services des réseaux sans fil vont permettre dans un futur proche le développement d'applications distribuées temps réel. Cependant, certains problèmes propres à ce type d'applications doivent avant tout être résolus de manière plus globale. On peut citer par exemple la gestion des flux de transmission, la définition d'un protocole application adapté aux réseaux sans fil pour la commande temps réel, la gestion de la pertinence des ordres de l'opérateur distant,

la définition d'une architecture matérielle et logicielle adéquate mais aussi la stabilité des systèmes bouclés soumis à un retard variable. Ces verrous scientifiques sont actuellement à l'étude et c'est à travers les premiers résultats obtenus dans chacun de ces domaines que nous avons pu présenter cet article qui représente la finalité applicative visée : l'utilisation de Bluetooth pour la commande à distance de robots mobiles.

Remerciements

Cette activité de recherche résulte d'une collaboration scientifique et financière L2I entre le LIRMM-CNRS de Montpellier, l'équipe Réseaux et Protocoles du LIMOS de Clermont-Ferrand, et l'équipe de recherche ICARE EA 3050 de Toulouse-Blagnac.

7. Bibliographie

- [802.15, 2002] 802.15. GROUPE DE TRAVAIL IEEE 802.15 SUR LES WPAN. 2002.
[HTTP://GROUPER.IEEE.ORG/GROUPS/802/15/](http://grouper.ieee.org/groups/802/15/).
- [ADVANTECH, 2002] ADVANTECH. ORDINATEURS INDUSTRIELS. 2002.
[HTTP://WWW.ADVANTECH.COM/EPC](http://www.advantech.com/epc).
- [ERICSSON, 2002] ERICSSON. PRODUITS BLUETOOTH. 2002.
[HTTP://WWW.ERICSSON.COM/MICROE](http://www.ericsson.com/microe).
- [FRAISSE *et al.*, 1999] PHILIPPE FRAISSE, DIDIER MARTY-DESSUS, , RAPHAEL PROTIÈRE. TRANSMISSION DE L'INFORMATION. *Ellipses*, 1999.
- [FRAISSE *et al.*, 2000] PHILIPPE FRAISSE, MICHEL MISSON, , THIERRY VAL. ETUDE DE LA COMMANDE DE MANIPULATEUR MOBILE VIA LA TECHNOLOGIE BLUETOOTH. 2000. COLLABORATION SCIENTIFIQUE CADRE LIRMM, LIMOS, ICARE.
- [FRIDMAN, LEVANT, 2002] L FRIDMAN, A LEVANT. HIGH ORDER SLIDING MODES. *Sliding Modes Control in Engineering*, p. 53–101, 2002.
- [HAARTSEN, 2000] J. HAARTSEN. AD-HOC RADIO CONNECTIVITY VIA THE BLUETOOTH SYSTEM. *IEEE Personal, indoor and mobile radio communications*, LONDRES, UK, SEPTEMBER 2000.
- [JACQUET *et al.*, 2000] FRÉRIQUE JACQUET, N HAKEM, MICHEL MISSON, , THIERRY VAL. LES RÉSEAUX LOCAUX SANS FIL : LE POINT SUR L'ÉMERGENCE D'UNE SOLUTION ALTERNATIVE. *Calculateurs Parallèles*, 2000.
- [KAINKA, BERNDT, 1999] B. KAINKA, H.J. BERNDT. LES INTERFACES DE MON PC SOUS WINDOWS. *Librairie Delphi et Visual Basic, Publitronic*, 1999.
- [LEE, GERLA, 2001] S.J. LEE, M. GERLA. DYNAMIC LOAD-AWARE ROUTING IN AD-HOC NETWORKS. *Proceedings of ICC'01*, HELSINKI, FINLAND, JUNE 2001.
- [PINEL, 2001] P PINEL. SÉCURITÉ DES DONNÉES DANS LE TRANSFERT. *Séminaire Jessica sur Bluetooth*, LAAS/CNRS, 2001.
- [SIG, 2002] SIG. BLUETOOTH SIG, MEMBER WEB SITE. 2002.
[HTTP://WWW.BLUETOOTH.ORG](http://www.bluetooth.org).

- [UTKIN, 1977] V UTKIN. VARIABLE STRUCTURE SYSTEMS WITH SLIDING MODES. *IEEE Transactions on Automatic Control*, AC-22 :212–222, 1977.
- [VAL , JUANOLLE, 2002] THIERRY VAL , GUY JUANOLLE. DÉVELOPPEMENT D’APPLICATIONS DE MÉTROLOGIE POUR LE WPAN BLUETOOTH. *Colloque Francophone sur l’Ingénierie des Protocoles*, MONTRÉAL, QUÉBEC, CANADA, MAI 2002.
- [VAL *et al.*, 2003] THIERRY VAL, FABRICE PEYRARD, , MICHEL MISSON. STUDY AND SIMULATION OF THE INFRARED WLAN IRDA : AN ALTERNATIVE TO THE RADIO. *Computer Communications, Special issue of Ubiquitous Computing*, À PARAÎTRE, 2003.

Soumis le : 20 janvier 2003

Accepté le : 3 octobre 2003